Programmeringsopgave II

Protokolsimulering

I denne opgave skal der laves et værktøj, der kan visualisere, hvordan en distribueret protokol forløber.

Opgaven tager udgangspunkt i et sensornetværk, men abstraherer fra de konkrete geometriske forhold. Opgaven giver en introduktion til simulering af dynamiske systemer.

Begreber

En *graf* G består af en mængde af *punkter* V og en mængde af *kanter* $E \subseteq V \times V$. Det antages i denne opgave, at kanterne er ikke-orienterede og at en kant mellem punkt a og punkt b er repræsenteret både ved $(a, b) \in E$ og $(b, a) \in E$. [Dvs. at $E = E^{-1}$.]

Et *(diskret) udlægning* af en graf $G = (V, E)$ er en afbildning $\sigma : V \to Z \times Z$, hvor Z angiver mængden af heltal. For et punkt a, angiver $(x, y) = \sigma(a)$ koordinaterne for a i et retvinklet koordinatsystem. Vi vil antage at σ er injektiv.

I denne opgave benyttes en graf til at repræsentere strukturen af et *netværk* af kommunikerende enheder (fx. sensorer), her kaldet *knuder*. Punkter repræsenterer knuder og kanter repræsenterer kommunikationsforbindelser.

Grafisk visning

En udlagt graf vises naturligt i et retvinklet koordinatsystem, hvor punkter afsættes på deres udlagte position og kanter tegnes som linier imellem dem.

Under simuleringen kan kommunikation over en forbindelse fx. vises ved farvning og tilstanden af en knude ligeledes ved farver, form eller tekst.

Opgave

Der skal laves værktøj til visning af et netværk af kommunikerende enheder (givet som en udlagt graf) og visualisering af distribuerede protokoller.

Værktøjet skal kunne:

- Indlæse en udlagt graf fra en fil og vise den.
- Kunne markere et eller flere punkter som *terminaler* (dvs. endestationer for data genereret af knuderne).
- Kunne simulere protokollen til rutebestemmelse. Protokollens forløb skal visualiseres på grafen. Simuleringsskridtene skal kunne drives manuelt, dvs. ved tryk på en knap/tast for hvert skridt. Den simulerede tid (antal skridt) skal vises.
- Gemme en simuleringstilstand i en fil og indlæse den igen

For grupper af 3 personer forventes mindst en af nedenstående udvidelser implementeret.

Om simulering

Ved simulering af et system skal man efterligne et forløb, hvor en række objekter i systemet udvikler sig og vekselvirker over et tidsrum. I dette tilfælde foregår vekselvirkningen ved kommunikation af beskeder.

En simpel form for simulering kan foretages ved at antage, at forløbet kan deles op i en række "tidsskridt" af samme længde, og så lade alle objekter ændre sig "samtidigt", ét skridt ad gangen. Fx kan man antage, at overførsel af en besked tager netop et tidsskridt. En driver-rutine for simuleringen kan så henvende sig til alle simuleringsobjekter (her netværkets knuder) en efter en og bede dem om at udføre (deres del af) et simuleringsskridt. Når alle simuleringsobjekter har udført hver deres del, er det samlede skridt udført og simuleringen er klar til udførelse af næste skridt.

For den konkrete rutningsprotokol, vil hver knude i netværket have en tilstand, der angiver den hidtil korteste vej til en terminal med tilhørende retning (en nabo). Såfremt en knude under et simuleringsskridt modtager en kortere vej, skal den selv *broadcaste* den nye længde til alle sine naboer i *næste* skridt. For at adskille kommunikationer hørende til forskellige skridt, skal skridtene udføres i to *faser*: En *forberedelsesfase*, hvor knuderne udveksler information baseret på deres nuværende tilstand, og en egentlig *transitionsfase* hvor knuderne på basis heraf går til en ny tilstand.

Hvert knude-objekt bør derfor have to metoder til understøttelse af simuleringen: En `prepare()` metode, hvor hver knude udsender beskeder til (nogle af) sine naboer og en `step()` metode, der bringer knuden i en ny tilstand på basis af den forrige tilstand og modtagne beskeder.

Beskeder kan fx. udveksles ved, at hver knude har en `receive(...)` metode hvor andre knuder kan aflevere beskeder under forberedelsesfasen. Bemærk at `receive()` kan blive kaldt både *før* og *efter* at knuden selv udsender beskeder. Beskeder modtaget i forberedelsesfasen skal derfor generelt blot samles op, fx. i en liste, til brug ved det næste kald af `step()`.

Vink

Det kan antages at grafer udlægges inden for et begrænset område, fx. 100×100.

Ved at anvende et simpelt tekstuelt format for udlagte grafer kan I manuelt lave grafer med en tekst-editor.

Rutningsprotokollen kan findes i materialet for efterårets Uge 10 [CampusNet].

Ideer til udvidelser

- Redigering af grafen: Flytning af punkter, tilføjelse/sletning af kanter, tilføjelse/sletning af punkter. Gemning af redigeret graf i fil.
- Automatisk simulering, dvs. at simuleringen kører af sig selv med en vis hastighed, efter at være blevet sat igang. Skal kunne stoppes igen.

Hertil bør man benytte Swing's `Timer` klasse, der kan generere "tik", der kan drive simuleringen (se mere på materialesiden). [Undgå brug af tråde (eng.*threads*).]

- Simulering af andre protokoller. Fx. positioneringsprotokollen.
- En mere realistisk simulering/protokol. Fx. ved at tillade at knuder går ned, at beskeder gå tabt ved med en vis sandsynlighed (evt. afhængig af afstand), eller ved at lade kommunikation og behandling tage varierende tid. Protokollen må så gøres tilsvarende mere robust, fx. ved at ruter opdateres regelmæssigt.
- Mulighed for at gå frem og tilbage i en simulering.